

Amendments to the claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1.(CURRENTLY AMENDED) A method ~~Method~~ to secure the execution of at least one ~~module~~ program in an electronic ~~unit~~ assembly comprising information processing means and information storage means, ~~characterised in that,~~ comprising
inserting directives, corresponding to beacons
and beacon functions intended for a pre-processor, into
the code of the program;
causing the pre-processor to replace at least one
directive by a beacon determined to correspond to the
directive, in the code of the program;
during the execution of said ~~module~~program, it
causing the electronic assembly to: consists,
during the passage by at least one beacon, in
storing store one or more items of information
concerning one or more characteristics of ~~said~~ at least
one beacon during the passage of said beacon and in
checkingcheck, at, at least one check point, the
consistency of the information stored about all beacons
encountered.

2. (CURRENTLY AMENDED) A method ~~Method~~ to secure at least one ~~module~~
program designed to be integrated in an electronic ~~unit~~ assembly including
information processing means and information storage means, ~~characterised in that it~~
~~integrates automatically in said module equipped with~~ comprising

inserting directives, corresponding to beacons and beacon functions, and intended for a pre-processor, into the code of the program;

causing the pre-processor to replace the directives in the code of the program a set of directives replaced by a set of static data; a set of static data, beacon functions and verification functions to automatically integrate a set of valid executions represented by the static data, the beacon functions being used for calculating dynamically a representation of the execution and the verification functions being used to check the consistency of the static and dynamic data.

~~the first representing a set of valid executions, the second calculating dynamically a representation of the execution, and the last used to check the consistency of the static and dynamic data.~~

3. (CURRENTLY AMENDED) The method Method according to claim 2, ~~characterised in that it~~ wherein the electronic unit uses ~~the a~~ control flow graph of the program to be protected to generate the static information used by the verification functions.

4. (CURRENTLY AMENDED) The method Method according to claim 2 or 3, ~~characterised in that~~ wherein a beacon is information which defines the characteristics of the corresponding passage point and/or one or more other passage points.

5. (CURRENTLY AMENDED) The method ~~Method~~ according to claim 4, ~~characterised in that~~ wherein a ~~the~~ beacon is one of the following elements, a combination of several of them, or all of them:

an integer locating the beacon in the code to be protected;

a Boolean variable defining whether ~~it~~ the ~~beacon~~ is the first or the last beacon;

a data structure characterizing, according to the value of a register or a given variable, all beacons through which passage will be forbidden (using a verification function) in the remaining execution;

a data structure ~~characterizing~~ comprising, according to the value of a register or a given variable, all beacons through which passage will be forced (using a verification function) in the remaining execution.

6. (CURRENTLY AMENDED) The method ~~Method~~ according to ~~one of~~ claims 2 or ~~3 to 5~~, ~~characterised in that~~ wherein a beacon function is ~~one which is~~ called by the program at each passage by a beacon and which ~~will consist~~ consists in storing dynamically in ~~the~~ a shared memory various items of information concerning the beacon.

7. (CURRENTLY AMENDED) The method ~~Method~~ according to claim 6, ~~characterised in that a~~ wherein the beacon function is ~~one which~~ pushes the beacon onto ~~the~~ a stack in the shared memory and/or one which updates a checksum contained in the shared memory with the beacon data.

8. (CURRENTLY AMENDED) The method ~~Method~~ according to one of claims 2 or 3 to 7, characterised in that further comprising calling a history verification function ~~is one called~~ at each check point to check the consistency of the information stored in the shared memory during the successive calls of the beacon functions.

9. (CURRENTLY AMENDED) An electronic ~~Electronic unit~~ assembly including information processing means and information storage means containing at least one ~~module~~ program to be executed ~~characterised in that it~~ wherein the electronic assembly comprises ~~includes~~ the means required, during the execution of said ~~module~~ program, and during the passage by at least one beacon, to store one or more items of information concerning one or more characteristics of said beacon in said storage means and means to check, at at least one check point, the consistency of the information stored about all beacons encountered.

10. (CANCEL) Program including program code instructions to execute the steps of the method according to one of claims 1 to 6 when said program is run in an electronic unit.

11. (NEW) A computer executable storage medium having a computer executable program code embodied therein, said computer executable program code adapted to be executed to secure the execution of at least one program in an electronic assembly having information processing means and information storage means, said executable program code comprising instructions to direct a microprocessor of the electronic assembly to:

cause a pre-processor to replace at least one directive inserted into a program, the at least one directive corresponding to beacon and beacon functions, and intended for the pre-processor, by a determined beacon in the code of the program; and

during the execution of said program:

store one or more items of information
concerning one or more characteristics of at least one
beacon during the passage by said beacon and

check at, at least one check point, the
consistency of the information stored about all beacons
encountered.

12. (NEW) A computer readable storage medium having a computer executable program code embodied therein, said computer executable program code adapted to be executed to secure the execution of at least one program in an electronic assembly having information processing means and information storage means, said executable program code comprising instructions to direct a microprocessor of the electronic assembly to:

to replace directives inserted into a program,
the directives corresponding to beacon and beacon
functions, by a set of static data, beacon functions and
verification functions to automatically integrate a set of
valid executions represented by the static data, the
beacon functions being used for calculating
dynamically a representation of the execution; and

the verification functions being used to check
the consistency of the static and dynamic data.

13. (NEW) The computer executable storage medium of Claim 12 wherein the medium further comprises instructions to direct a microprocessor of the electronic assembly to use a control flow graph of the program to be protected to generate the static information used by the verification functions.

14. (NEW) The computer executable storage medium of Claim 12 or 13 wherein a beacon is information which defines the characteristics of the corresponding passage point and/or one or more other passage points.
15. (NEW) The microprocessor module of Claim 14 wherein the beacon is one of the following elements, a combination of several of them, or all of them:
 - an integer locating the beacon in the code to be protected;
 - a Boolean variable defining whether the beacon is the first or the last beacon;
 - a data structure characterizing, according to the value of a register or a given variable, all beacons through which passage will be forbidden (using a verification function) in the remaining execution;
 - a data structure comprising, according to the value of a register or a given variable, all beacons through which passage will be forced (using a verification function) in the remaining execution.
16. (NEW) The microprocessor module of Claim 12 wherein the beacon function is called by the module at each passage by a beacon and which will consist in storing dynamically in a shared memory various items of information concerning the beacon.